LOW COST, HIGH EFFICIENT MULTI TRANSFORM CORE SUPPORTING HIGH RESOLUTION VIDEO CODECS USING COMMON SHARING DISTRIBUTED ARITHMETIC

V SANTHOSH KUMAR, NAZIA
Associate Professor, Assistant Professor
Syed Hashim College of Science and Tech

ABSTRACT
An efficient architecture of 1D CSDA MST core is designed using CSDA (Common Sharing Distributed Arithmetic) to achieve high-throughput rate supporting multistandard transformations at low cost. Common sharing distributed arithmetic (CSDA) combines factor sharing and distributed arithmetic sharing techniques, efficiently reducing the number of adders for high hardware-sharing capability. Conventional distributed arithmetic (DA) is popular in application specific integrated circuit (ASIC) design, and it features on-chip ROM to achieve high speed and regularity. In this paper, a new DA architecture called NEDA is proposed, aimed at reducing the cost metrics of power and area while maintaining high speed and accuracy in digital signal processing (DSP) applications. Compared with other architectures available, it has a great improvement on computing speed and reducing area. And also, an CLA (carry look ahead adder) is implemented and to achieve low power and high throughput discrete cosine transform (DCT) design. While conventional approaches use the original DCT algorithm, the proposed architecture uses the recursive DCT algorithm and requires less area than the conventional approaches, regardless of the memory reduction techniques employed in the ROM Accumulators (RACs). The main strategy aims to reduce the nonzero elements using CSDA algorithm and hence few adders are required in the adder-tree circuit.

Index terms: Discrete cosine transform (DCT), Factor sharing , Distributed arithmetic, 1-D MST, CLA.

1. Introduction
Multimedia communications such as image and video require high volume of data transmission. Data compression reduces the communication cost in transmitting data over long links because of bandwidth reduction. Discrete Cosine Transform (DCT) is an important component of image and video compression and is adopted on various standardized coding schemes, such as VC-1, MPEG and H.264. There are two main reasons for its importance: first, it is effective at transforming image data into a form that is easy to compress and secondly it can be efficiently implemented in software and hardware. Moreover, it has two useful properties for image and video compression, energy compaction (concentrating the image energy into a small number of coefficients) and de-correlation (minimizing the interdependencies between coefficients). Some applications require real-time manipulation of digital images, so many fast algorithms and specific circuits for DCT have been developed. Most of these applications are computation intensive with multiplier and/or adder being the predominant operator. An 8x8 parallel architecture is developed for real-time requirement. Its input data width is 8-bit, and the output coefficients are set to 13-bit width, which is accurate enough for the transformation and will not cause obvious image reconstructing distort.

1.1 Discrete cosine transform
DCT expresses a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies. It is a Fourier related transform similar to DFT. DCTs are equivalent to DFTs of roughly twice the length, operating on real data with even symmetry, where is some variants the input/output data shifted by half a sample. Like other transforms, the Discrete Cosine Transform (DCT) attempts to decorrelate the image data.

After decorrelation each transform Discrete cosine transform (DCT) is a key coding tool for video compression. It achieves data compression by converting the high relative spatial domain data into low relative frequent domain data. DCT is a computation intensive operation. It requires a large number of adders and multipliers for direct 2implementation. Multipliers consume more power and hence distributed arithmetic (DA) is used to implement multiplication without multiplier.

1.2 The One-Dimensional DCT
The most common DCT definition of a 1-
D sequence of length N is
\[ C(u) = \alpha(u) \sum_{x=0}^{N-1} f(x) \cos \left[ \frac{\pi}{N} (2x+1) \right] \]
for \( u = 0, 1, 2, \ldots, N-1 \). Similarly, the inverse transform is defined as,
\[ \sum_{x=0}^{N-1} \frac{\alpha(u) c(u) \cos \left[ \frac{\pi}{N} (2x+1) \right] }{N} \]
for \( x = 0, 1, 2, \ldots, N-1 \), in both equations (1) and (2), \( \alpha(u) = \frac{1}{N} \) for \( u = 0 \) and \( \alpha(u) = \sqrt{2} N \) for \( u \neq 0 \).

If the input sequence has more than \( N \) sample points then it can be divided into sub-sequences of length \( N \) and DCT can be applied to these chunks independently. Here, a very important point to note is that in each such computation the values of the basis function points will not change. Only the values of \( f(x) \) will change in each sub-sequence. This is a very important property, since it shows that the basis functions can be pre-computed offline and then multiplied with the sub-sequences. This reduces the number of mathematical operations (i.e., multiplications and additions) thereby rendering computation efficiency.

1.3 The Two-Dimensional DCT
The 2-D DCT is a direct extension of the 1-D case and is given by
\[ C(u,v) = \alpha(u) \alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \cos \left[ \frac{\pi}{N} (2x+1) \right] \cos \left[ \frac{\pi}{N} (2y+1) \right] \]
for \( u, v = 0, 1, 2, \ldots, N-1 \).

For \( x, y = 0, 1, 2, \ldots, N-1 \). The 2-D basis functions can be generated by multiplying the horizontally oriented 1-D basis function with vertically oriented set of the same functions. The basis functions for \( N = 8 \) are shown in. Again, it can be noted that the basis functions exhibit progressive increase in frequency both in the vertical and horizontal direction.

1.4 Factor Sharing
The FS method shares the same factor in different coefficients among the same input. Consider two different elements \( S1 \) and \( S2 \) with the same input \( X \) as an example
\[ S1 = C1X, \]
\[ S2 = C2X \]
Assuming that the same factor \( Fs \) can be found in the coefficients \( C1 \) and \( C2 \), (1) can be rewritten as follows:
\[ S1 = (Fs 2k1 + Fd1) X \]
\[ S2 = (Fs 2k2 + Fd2) X \]
where \( k1 \) and \( k2 \) indicate the weight position of the shared factor \( Fs \) in \( C1 \) and \( C2 \), respectively. \( Fd1 \) and \( Fd2 \) denote the remainder coefficients after extracting the shared factor \( Fs \) for \( C1 \) and \( C2 \), respectively
\[ Fd1 = C1 - Fs 2k1 \]
\[ Fd2 = C2 - Fs 2k2 \]

1.5 Canonical Signed Digit (CSD) Representation
The Canonical Signed Digit (CSD) number system is a signed digit number system that minimizes the number of non-zero digits and thus can reduce the number of partial product additions in a hardware multiplier. The encoding scheme uses a digit set that is ternary and each digit can be either -1, 0, or +1. Adjacent CSD digits are never both non-zero, i.e., \( c1 \cdot c2 = 0 \). This property implies that for an n-bit number, there are at most \( 2^n \) digits. For a 2” complement number, there can be \( n \) non-zero digits for an n-bit number. The inner product for a general matrix multiplication and accumulation can be written as follows
\[ Y = A^T X = \sum_{i=0}^{n} AiX_i \]

Properties of CSD Numbers
The following are the properties of CSD numbers:
- 2 consecutive bits in a CSD number are non-zero.
- The CSD representation of a number contains the minimum possible number of non-zero bits, thus the name canonical.

1.5 Common Sharing Distributed Arithmetic (CSDA) Algorithm
The proposed CSDA algorithm combines the FS and DA methods. By expanding the coefficients matrix at the bit level, the FS method first shares the same factor in each coefficient; the DA method is then applied to share the same combination of the input among each coefficient position. An example of the proposed CSDA algorithm in a matrix inner product is as follows
\[ \begin{bmatrix} Y1 \\ Y2 \end{bmatrix} = \begin{bmatrix} C11 & C12 \\ C21 & C22 \end{bmatrix} \begin{bmatrix} X1 \\ X2 \end{bmatrix} \]
the previous data (adder) and update the new data to adopt the searching flow in the iteration core software code will help to do the iterative searching loops by setting a constraint with minimum nonzero elements.
2.1 SELECTED BUTTERFLY MODULE

Selected butterfly model architecture performs the 8 point butterfly model with 8 multiplexers. These 8 point transform is divided into two 4 point transforms, like even part CSDA and odd part CSDA.

2.2 EVEN PART CSDA CIRCUIT

The input is the multistandard input signal which can be MPEG, H.264, VC-1 from which input matrix are extracted using XILINX and is given as input to the 1D CSDA MST core which consists of a selected butterfly (SBF) module, an even part CSDA (CSDA_E), an odd part CSDA (CSDA_O), eight error-compensated error trees (ECATs) to produce the transformed data. Iteration core in the CSDA circuit is used to find the CSDA shared coefficient using DA method based on FS method in order to reduce the number of adder and also to achieve high sharing capability and finally it will compare it to the CSDA even part calculates the even part of the 8 point transform similar to the four-point transform for H.264 and VC-1 standards. Here it consists of the two pipeline stage architectures, which is developed by using the D flipflops as pipeline registers. First stage executes the 4 point input butterfly matrix circuit and the second stage of even part CSDA tells about the hardware resources in variable video applications by using the proposed CSDA algorithm. Selection signals of multiplexers (MUXs) for different standards for CSDA.

2.3 ODD PART CSDA CIRCUIT

Similar to the even part CSDA, the odd part CSDA also consists of the two pipeline stage registers, which efficiently shares the hardware resources among the odd part of the architecture in variable video standard applications.

Conclusion

In this paper Conventional distributed arithmetic (DA) is popular in application specific integrated circuit (ASIC) design, and it features on-chip ROM to achieve high speed and regularity. In this paper, a new DA architecture called NEDA is proposed, aimed at reducing the cost metrics of power and area while maintaining high speed and accuracy in digital signal processing (DSP) applications. Compared with other architectures available, it has a great improvement on computing speed and reducing area. And also, an CLA (carry look ahead adder) is implemented and to achieve low power and high throughput discrete cosine transform (DCT) design. While conventional approaches use the original DCT algorithm, the proposed architecture uses the recursive DCT algorithm and requires less area than the conventional approaches, regardless of the memory reduction techniques employed in the ROM Accumulators (RACs). The main strategy aims to reduce the nonzero elements using CSDA algorithm and hence few adders are required in the adder-tree circuit.

REFERENCES


[5] PENG Chungan, CAO Xixin, YU Dunshan and ZHANG Xing, „A 250mhz Optimized Distributed Architecture of 2D 8x8 DCT,” SOC Lab., Peking University, Beijing, 100871, P.R.China April 2007.

